B9

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 751 462 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
31.07.2002 Bulletin 2002/31

(51) Int Cl.[7]: **G06F 11/14**

(21) Application number: **96302962.4**

(22) Date of filing: **26.04.1996**

(54) **A recoverable disk control system with a non-volatile memory**

Wiederherstellbares Plattensteuersystem mit nichtflüchtigem Speicher

Système de commande de disque restaurable avec mémoire rémanente

(84) Designated Contracting States:
DE FR GB

(30) Priority: **19.06.1995 JP 15173795**

(43) Date of publication of application:
02.01.1997 Bulletin 1997/01

(73) Proprietor: **KABUSHIKI KAISHA TOSHIBA**
Kawasaki-shi, Kanagawa-ken 210-8572 (JP)

(72) Inventors:
- **Shimizu, Kuniyasu, c/o Toshiba Corp.**
Tokyo (JP)
- **Hirayama, Hideaki, c/o Toshiba Corp.**
Tokyo (JP)

(74) Representative: **Shindler, Nigel**
Brookes Batchellor
102-108 Clerkenwell Road
London EC1M 5SA (GB)

(56) References cited:
EP-A- 0 572 019
US-A- 5 241 672
US-A- 5 392 445

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

EP 0 751 462 B1

Description

[0001] This invention relates to a computer system using a checkpoint method, and in particular, concerns a disk control system that enables high-speed disk access to avoid processing delay associated with the checkpoint method.

RELATED BACKGROUND ART

[0002] A computer system using the checkpoint method postpones write operation to a disk device until the next checkpoint is acquired. Otherwise, if the next time the computer system restarts from the last checkpoint because of a fault, the disk device would not be able to be restored to the state of the time of the last checkpoint. This causes a considerable performance degradation.

TECHNICAL PROBLEM AND THE SOLUTION

[0003] The purpose of this invention is to provide a recoverable disk device without the processing delay associated with the checkpoint method.

[0004] Another purpose of this invention is to accelerate a disk write operation by using a nonvolatile memory.

[0005] In accordance with the present invention, the foregoing objects, among others, are achieved by providing a computer system executing a checkpoint operation, the system restarting from the last checkpoint status if a fault occurred, the system characterized in that said disk device is for reading and storing data, said operating system is for issuing a write or a read request to the disk device and in that the system further includes a nonvolatile memory, write request storage means for storing write data and an associated write request in the nonvolatile memory, and disk write means for storing the write data stored in the nonvolatile memory by the write request storage means into the disk device, the disk write means further including means for storing write data according to the write request generated before the last checkpoint from the nonvolatile memory to the disk device.

[0006] There has also been provided, in accordance with another aspect of the present invention, a method of controlling disk access in a computer system executing a checkpoint operation periodically, the system restarting from the last checkpoint prior to a fault occuring therein, the system comprising a disk device for storing data to be processed in the system and an operating system for issuing a write request to the disk device, characterized in comprising the steps of first storing write data and an associated write request from said operating system to said disk device into a nonvolatile memory, second storing the write data stored in the nonvolatile memory into said disk device, the second storing step including the step of storing write data generated before the last checkpoint from the nonvolatile memory to the disk device.

[0007] This invention stores write data requested by an operating system into the nonvolatile memory. Also this invention stores write data corresponding to a write request generated before an execution of the last checkpoint from the nonvolatile memory to the disk device.

[0008] This invention can secure fault resistance. The disk control system of this invention can improve the latency of the disk access by avoiding the delay of the processing resulting from view of checkpoint execution.

[0009] Other objects, features, and advantages of the present invention will become apparent from the following detailed description. It should be understood, however, that the detailed description and specific examples, while indicating preferred embodiments of the invention, are given by way of illustration only, since various changes and modifications within the scope of the invention will become apparent to those skilled in the art from this detailed description.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] Figure 1 is a schematic block diagram of a computer system using a checkpoint and in which the disk control system of this invention is applied.

[0011] Figure 2 is a schematic block diagram illustrating the elements of one embodiment of a computer system where a write request sends data to a nonvolatile memory.

[0012] Figure 3 is a schematic block diagram illustrating the movement of data from the indefinite queue to the definite queue.

[0013] Figure 4(a) is a flow chart illustrating the write process of this disk control system.

[0014] Figure 4(b) is a flow chart illustrating the checkpoint operation of this disk control system.

[0015] Figure 4(c) is a flow chart illustrating the background processing of this disk control system.

[0016] Figure 5(a) is a block diagram illustrating the link control structure of a nonvolatile memory.

[0017] Figure 5(b) is a flow chart illustrating the processing of a disk write.

[0018] Figure 5(c) illustrates write data and write request structures in the nonvolatile memory.

[0019] Figures 6 and 7 are block diagrams illustrating the operation of a disk control system, when a fault occurs in the computer system.

[0020] Figure 8 is a time chart illustrating the operation of a disk control system, when a fault occurs in the computer system.

[0021] Figure 9(a) is a flow chart illustrating the recovery sequence (1) of this disk control system.

[0022] Figure 9(b) is a flow chart illustrating the recovery sequence (2) of this disk control system.

[0023] Figure 9(c) is a flow chart illustrating the read processing of this disk control system.

DESCRIPTION OF SPECIAL EMBODIMENT

[0024] One embodiment of this invention is explained, referring to the drawings.

[0025] Figure 1(a) is a block diagram of a computer system using a checkpoint method to which the disk control system of this invention is applied.

[0026] When a processor 1 updates the content of a main memory 2, this computer system records the before image of the updated portion into a log memory 6. The information of context on the processor and the content of a cache memory are written in the main memory at a checkpoint. The state that is needed to restart data processing from the last checkpoint is kept in the main memory and the log memory. When a fault occurs, the computer system restores the main memory state of the last checkpoint by writing back the before image to the updated portion of the main memory. At the end of the checkpoint operation, the content of the log memory is cleared. Moreover, the computer system of this invention has a nonvolatile memory 3 as an auxiliary memory and a disk device 4.

[0027] Figure 1(b) is a block diagram of the disk control system of this invention. As shown in Figure 1(b), the disk control system has a write request storage means 11, a write request fix means 12, a write request discard means 13 and a disk write means 14.

[0028] When an operating system generates a write request to the disk device, the write request storage means stores the write request and the associated write data into the nonvolatile memory. Although the write request is kept in the nonvolatile memory at this point, the associated data can not be written to the disk device before the next checkpoint. If a fault occurs before the end of next checkpoint operation, the write request and write data are discarded. Hereafter, we say that such a write request and data stored in the nonvolatile memory are in an indefinite state.

[0029] When the operation of the next checkpoint is executed without any faults, the write request fix means changes the state of the write request in the nonvolatile memory from an indefinite state to a definite state. This means that even if a fault occurs afterward such a change in state, the write request is definitely to be executed.

[0030] If a fault occurs before the next checkpoint execution, the write request discard means discards disk write requests and the associated data in the indefinite state.

[0031] The disk write means executes a disk write operation according to disk write requests in the definite state kept in the nonvolatile memory.

[0032] Next, the operation of this invention is explained, referring to Figures 2 - 9, which are block diagrams and flow charts illustrating the operation of one embodiment of this invention.

[0033] First, as shown in Figure 1(a), when a write request to the disk device is received from the operating system, the write request storage means puts the write request and the associated write data in the nonvolatile memory.

[0034] As depicted in Figure 2, the nonvolatile memory has an indefinite queue 31 and a definite queue 32. A write request and the associated write data received after the last checkpoint are stored in the indefinite queue of the nonvolatile memory. Upon execution of the checkpoint without a fault, the write requests and the associated write data in the indefinite queue are moved to the definite queue of the nonvolatile memory.

[0035] When a write request is stored in the indefinite queue of the nonvolatile memory by the write request storage means, the disk control system informs "write completion" to the operating system (S13 in Figure 4 (a)). Thereby, the operating system does not have to wait for the actual completion of the write request, which improves performance considerably.

[0036] Next, the operation of the disk control system during a checkpoint execution is explained, referring to Figure 3.

[0037] It is assumed that the checkpoint of this computer system is executed at a state shown in Figure 2. When the nonvolatile memory is filled with write data, a checkpoint is compulsorily executed. The disk control system checks the size of the free area of the nonvolatile memory. If the capacity of the nonvolatile memory is below a predetermined value, or after a fixed time to execute the checkpoint of the computer system has elapsed, I/O processing from the operating system is restarted. The I/O processing from the operating system is delayed, while the checkpoint is being executed.

[0038] During checkpoint operation, first, write requests and their write data stored in the indefinite queue are moved to the definite queue by the write request fix means (Figure 3 and S15 in Figure 4(b)). This movement does not necessarily need physical movement of the whole data but changing pointers. The method of controlling a write request and write data stored in the nonvolatile memory is explained later, referring to Figure 5.

[0039] After the checkpoint, the disk write means executes the disk write operation according to the write request stored in the definite queue (Figure 3 and S17 in Figure 4(c)). It is recommended that the write operation to the disk device by the disk write means is properly executed according to the priority of the write request and the load of the computer system.

[0040] The nonvolatile memory contains an indefinite queue, a definite queue, request structures, and write data, referring to Figure 5(a). The indefinite queue and the definite queue have a control block (not shown) for managing the request structures. The indefinite queue has two request structures 4 and 5 as shown in Figure 5(a). The definite queue has .three request structures 1, 2 and 3 as shown in Figure 5(a). Control information on a request structure has a write flag which shows that the data is now writing to the disk device, pointer infor-

mation to the write data and next request structure, a disk number N, a sector number M, and the size of write data L, to store the write data kept in the definite queue into the disk device as shown in Figure 5(c). The size of the write data is a multiple of 512 bytes. For instance, the request structure 3 has the write data of 2*512 bytes.

[0041] When a next checkpoint operation is executed, the next request pointer information of the request structure 3 is changed to point a request structure 4. The next request pointer information of a request structure 5 remains unchanged. The control block of the indefinite queue also has means for managing the request structure 4 until a next checkpoint operation is executed. The control block of the indefinite queue does not manage the request structure 4 after the next checkpoint operation is executed.

[0042] Next, a write operation to the disk is explained referring to Figure 3. The disk write means searches the define queue for the request structure where a write flag is not set yet (S100 in Figure 5(b)). The disk write means gets the number of the request structures, the size of the write data, and the control information on the request structure from the control block one by one. Next, the disk write means sets the write flag of the request structure 1, of which the flag has not been set yet, through the control block of the definite queue (S110 in Figure 5 (b)). The disk write means stores the write data in the disk according to the request structure 1 (S130 in Figure 5(b)). When the writing process of the disk ends, the disk write means removes the request structure 1 and the associated data from the link through the control block (S150 in Figure 5(b)). The write flags of the request structures 2, 3, 4 and 5 are set one by one by the similar way. The write data of the request structures 2, 3, 4 and 5 are written in the disk device.

[0043] Here, the operation of this invention is explained, when a fault occurs in the computer system, referring to Figures 6 - 8.

[0044] As shown in Figure 8, a disk write request from the operating system stores write requests and write data A, B and C in the indefinite queue in the nonvolatile memory one by one (point ① in Figure 8). The operating system reads the data C from the indefinite queue in the nonvolatile memory (point ② in Figure 8). During the next checkpoint operation as shown in Figure 8, under such a condition, A, B, and C that are stored in an indefinite queue of the nonvolatile memory, link to a definite queue (CKP point in Figure 8). Also, a disk write request and write data from the operating system generated after the last checkpoint are linked to the indefinite queue.

[0045] After that, the write request and the write data D, E and F are linked in the indefinite queue of the nonvolatile memory one by one (point ③ in Figure 8). A disk read request from the operating system reads the data F from the indefinite queue and the data C from the definite queue in the nonvolatile memory (point ④ in Figure 8)

[0046] It is assumed that a fault occurred in the computer system after the checkpoint (breakdown point in Figure 8). The three write requests of 6d-6f (D, E and F) were stored in the indefinite queue as shown in Figures 6 and 8. The three write requests of 6a-6c (A, B and C) were stored in definite queue as shown in Figures 6 and 8.

[0047] In this case, the computer system of this invention first restores the computer system to the state of the last checkpoint (CKP point in Figure 8). The computer system begins restarting from the checkpoint state after the restoring process.

[0048] When the computer system is restarted, the disk control system discards all the write requests 6d-6f stored in the indefinite queue in the nonvolatile memory by the write request discard means (S17 in Figure 9(a)). The write requests 6d-6f and the associated data are issued again because the system restarts from the last checkpoint state as mentioned above.

[0049] Figure 7 illustrates the state of the nonvolatile memory after this recovery processing.

[0050] On the other hand, even if the state of the computer system rolls back to the last checkpoint, the write requests 6a-6c stored in the definite queue in the nonvolatile memory are effective. Therefore the disk write requests 6a-6c are properly stored in the disk device by the disk write means (S19 in Figure 9(b)).

[0051] As a result consistency of disk operations is maintained under control of the checkpoint method with this recoverable disk control system. In addition, the delay of the disk write according to the checkpoint method can be excluded. Thus, the latency of the disk access can be improved.

[0052] Next, a read request from the operating system is explained, referring to Figures 8 and 9.

[0053] At this time, the disk control system decides whether there are data to be read from the nonvolatile memory (S21 in Figure 9(c)). If there are data to be read out (Yes of S21 in Figure 9(c)), the data is read from the nonvolatile memory (② and ④ points in Figures 8 and S25 in Figure 9(c)). After that, this read processing ends.

[0054] On the other hand, if there are not data to be read from the nonvolatile memory (No of S21 in Figure 9(c)), the disk control system accesses the disk device (S23 in Figure 9(c)). At this point, it is recommended that the data read from the disk device is able to be stored in the nonvolatile memory.

[0055] As explained in detail above, when a fault in the computer system occurs, this invention may have stored a write request into the definite queue of the nonvolatile memory. The write request which is generated before the last checkpoint operation is executed. Also this invention stores a write request in the indefinite queue of the nonvolatile memory, the write request which is generated after the last checkpoint is executed after the next checkpoint. For instance, even if a power supply failure occurs in the computer system, the information of the write request is maintained because the

information on the write request is stored in the nonvolatile memory.

[0056] The disk control system of this invention abandons all the write requests and the write data generated after the last checkpoint which are stored in the indefinite queue in the nonvolatile memory. However, the data in the definite queue is written into the disk device. Thus, even if the computer system is broken down by the power supply failure, the write data in the definite queue is written into the disk device at the reboot of the computer system.

[0057] Concurrently with the processing of write requests from the operating system, write data which are generated before the last checkpoint is stored in the disk device.

[0058] Moreover, as for a read request to the disk device, the disk control system of this invention first examines whether the requested data is on the nonvolatile memory so that the read request keeps coherency just like normal access to the disk device.

[0059] As a result, the disk control system of this invention can secure fault tolerance of the computer system. The disk control system of this invention can improve the latency of the disk access by avoiding the delay resulting from a checkpoint execution.

[0060] Numerous modifications and variations of the present invention are possible in light of the above teachings. It is therefore to be understood that, within the scope of the appended claims, the present invention can be practiced in a manner other than as specifically described herein.

Claims

1. A computer system executing a checkpoint operation periodically, the system restarting from the last checkpoint prior to a fault occurring therein, the system comprising a disk device (4) for storing data to be processed in the system, and an operating system (15) for issuing a write request to the disk device, **characterized in that**:

   said disk device is for reading and storing data;
   said operating system is for issuing a write or a read request to the disk device; and **in that** the system further includes

   a nonvolatile memory (3);
   write request storage means (11) for storing write data and an associated the write request in the nonvolatile memory; and
   disk write means (14) for storing the write data stored in the nonvolatile memory by the write request storage means into the disk device, the disk write means further including means for storing write data according to the write request generated before the last checkpoint from the nonvolatile memory to the disk device.

2. The computer system of claim 1, **characterized in that** further the system includes means for notifying a write completion to the operating system, after storing the write request and the associated write data in the nonvolatile memory, when a write request from the operating system to the disk device is issued.

3. A computer system of claim 1, **characterized in that** further the system includes:

   means for reading the write data from the nonvolatile memory if there is the corresponding data on the nonvolatile memory, when the operating system issues a read request to the disk device; and
   means for reading the write data from the disk device, if there is not the corresponding data on the nonvolatile memory.

4. The computer system of claim 1, **characterized in that** further the system includes write request discard means (13) for invalidating write data and corresponding write requests generated after the last checkpoint, when a fault occurred in the system.

5. The computer system of claim 1, **characterized in that** the write request storage means includes an indefinite queue (31) for storing a write request which is generated after the last checkpoint, and a definite queue (32) for storing a write request which is generated before the last checkpoint.

6. The computer system of claim 5, **characterized in that** further the system includes write request fix means (12) for moving the write data stored in the indefinite queue to the definite queue, while a checkpoint operation is executed.

7. The computer system of claim 5, **characterized in that** the write request storage means includes means for storing the write request and the associated write data in the indefinite queue.

8. The computer system of claim 5, **characterized in that** the disk write means includes means for storing the write data stored in the definite queue into the disk device.

9. The computer system of claim 5, **characterized in that** the system includes means for storing all write data stored in the definite queue into the disk device, while rebooting the computer system.

10. The computer system of claim 5, **characterized in**

that the indefinite and definite queues includes control block means for controlling each of write data and the associated write request, each of the write data having a request structure for linking write data and next request structure.

11. The computer system of claim 10, **characterized in that** the request structure includes a writing flag for indicating that the write processing to the disk device is executed.

12. The computer system of claim 11, **characterized in that** the disk write means comprises:

means for detecting whether a write flag of the request structure is set or not;
means for setting the writing flag, if the write flag is not set;
means for storing write data of the definite queue to the disk device, based on information of the request structure; and
means for removing the request structure and the associated write data when writing the data to the disk device is completed.

13. A method of controlling disk access in a computer system executing a checkpoint operation periodically, the system restarting from the last checkpoint prior to a fault occuring therein, the system comprising a disk device for storing data to be processed in the system and an operating system for issuing a write request to the disk device, **characterized in** comprising the steps of:

first storing (S10) write data and an associated write request from said operating system to said disk device into a nonvolatile memory (3);
second storing (S17, S130) the write data stored in the nonvolatile memory (3) into said disk device (4), the second storing step including the step of storing write data generated before the last checkpoint from the nonvolatile memory (3) to the disk device (4).

14. The method of claim 13, further comprising the step of notifying (S13) a write completion to the operating system, after storing the write data and the associated write request to the nonvolatile memory, when a write request from the operating system to the disk device is issued.

15. The method of claim 13, further comprising the steps of:

first reading (S21, S25) the write data from the nonvolatile memory if there is acquired write data among write data stored in the nonvolatile memory, when the operating system issues a read request to the disk device ; and
second reading (S21, S23) write data from the disk device, if there is not acquired write data among write data stored in the nonvolatile memory (3).

16. The method of claim 13, further comprising the step of invalidating (S17) all write data and the corresponding write request generated after the last checkpoint, if a fault has occurred.

17. The method of claim 13, wherein the nonvolatile memory includes an indefinite queue (31) for storing write data and the associated write request which generated after the last checkpoint, and a definite queue (32) for storing write data to the disk device generated before a last checkpoint, further comprising the step of moving (S15) the write request and write data stored in an indefinite queue to a definite queue, while a checkpoint operation is executing.

18. The method of claim 13, wherein the nonvolatile memory includes an indefinite queue (31) for storing write data and the associated write request which generated after the last checkpoint, and a definite queue (32) for storing write data into the disk device generated before the last checkpoint, wherein the second storing step includes the step of storing (S17) the write data stored in the definite queue into the disk device.

19. The method of claim 13, wherein the nonvolatile memory includes an indefinite queue (31) for storing write data and the associated write request which generated after the last checkpoint, and a definite (32) queue for storing write data and the associated write request to the disk device generated before the last checkpoint, further comprising the step of storing (S19) all write data stored in the definite queue into the disk device, while recovering the computer system.

20. The method of claim 13, wherein the nonvolatile memory includes an indefinite queue (31) for storing write data and the associated write request which generated after the last checkpoint, wherein the first storing step includes the step of discarding (S17) the write data and write requests in the indefinite queue after a fault occurred.

Patentansprüche

1. Ein Computersystem, das periodisch eine Prüfpunktoperation ausführt, wobei das System von dem letzten Prüfpunkt, der vor einem Fehler darin auftritt, neu startet, wobei das System eine Platten-

einrichtung (4) zum Speichern von in dem System zu verarbeitenden Daten und ein Betriebssystem (15) zum Ausgeben einer Schreibanfrage an die Platteneinrichtung aufweist, **dadurch gekennzeichnet, daß**

die genannte Platteneinrichtung zum Lesen und Speichern von Daten vorgesehen ist;
das genannte Betriebssystem zum Ausgeben einer Schreib- oder einer Leseanfrage an die Platteneinrichtung vorgesehen ist und das System ferner aufweist:

einen nichtflüchtigen Speicher (3);
Schreibanfragenspeichermittel (11) zum Speichern von Schreibdaten und einer zugehörigen Schreibanfrage in den nichtflüchtigen Speicher; und
Plattenschreibmittel (14) zum Speichern der in dem nichtflüchtigen Speicher gespeicherten Daten durch die Schreibanfragenspeichermittel in die Platteneinrichtung, wobei die Plattenschreibmittel ferner Mittel zum Speichern von Schreibdaten entsprechend der Schreibanfrage aufweisen, die vor dem letzten Prüfpunkt von dem nichtflüchtigen Speicher zu der Platteneinrichtung erzeugt worden ist.

2. Das Computersystem nach Anspruch 1, **dadurch gekennzeichnet, daß** das System ferner Mittel aufweist zum Melden einer Beendigung des Schreibens an das Betriebssystem nach dem Speichern der Schreibanfrage und der zugehörigen Schreibdaten in den nichtflüchtigen Speicher, wenn eine Schreibanfrage von dem Betriebssystem an die Platteneinrichtung ausgegeben wird.

3. Ein Computersystem nach Anspruch 1, **dadurch gekennzeichnet, daß** das System ferner aufweist:

Mittel zum Lesen der Schreibdaten aus dem nichtflüchtigen Speicher, falls sich die entsprechenden Daten in dem nichtflüchtigen Speicher befinden, wenn das Betriebssystem eine Leseanfrage an die Platteneinrichtung ausgibt; und
Mittel zum Lesen der Schreibdaten aus der Platteneinrichtung, falls sich die entsprechenden Daten nicht in dem nichtflüchtigen Speicher befinden.

4. Das Computersystem nach Anspruch 1, **dadurch gekennzeichnet, daß** das System ferner Schreibanfragenverwerfmittel (13) aufweist zum Ungültigmachen der Schreibdaten und zugehöriger Schreibanfragen, die nach dem letzten Prüfpunkt erzeugt wurden, wenn ein Fehler in dem System aufgetreten ist.

5. Das Computersystem nach Anspruch 1, **dadurch gekennzeichnet, daß** das Schreibanfragenspeichermittel eine unbestimmte Warteschlange (31) zum Speichern einer Schreibanfrage, die nach dem letzten Prüfpunkt erzeugt wurde, und eine bestimmte Warteschlange (32) zum Speichern einer Schreibanfrage, die vor dem letzten Prüfpunkt erzeugt wurde, aufweist.

6. Das Computersystem nach Anspruch 5, **dadurch gekennzeichnet, daß** das System ferner Schreibanfragenfixiermittel (12) aufweist zum Verschieben der in der unbestimmten Warteschlange gespeicherten Schreibdaten in die bestimmte Warteschlange, während eine Prüfpunktoperation ausgeführt wird.

7. Das Computersystem nach Anspruch 5, **dadurch gekennzeichnet, daß** das Schreibanfragenspeichermittel Mittel zum Speichern der Schreibanfrage und der zugehörigen Schreibdaten in der unbestimmten Warteschlange aufweist.

8. Das Computersystem nach Anspruch 5, **dadurch gekennzeichnet, daß** das Plattenschreibmittel Mittel zum Speichern der in der bestimmten Warteschlange gespeicherten Schreibdaten in der Platteneinrichtung aufweist.

9. Das Computersystem nach Anspruch 5, **dadurch gekennzeichnet, daß** das System Mittel aufweist, die alle in der bestimmten Warteschlange gespeicherten Schreibdaten in die Platteneinrichtung schreibt en, während das Computersystem neu bootet.

10. Das Computersystem nach Anspruch 5, **dadurch gekennzeichnet, daß** die unbestimmte und die bestimmte Warteschlange Steuerblockmittel aufweisen zum Steuern jeweils der Schreibdaten und der zugehörigen Schreibanfrage, wobei die Schreibdaten jeweils eine Anfragestruktur zum Verbinden der Schreibdaten und der nächsten Anfragestruktur aufweisen.

11. Das Computersystem nach Anspruch 10, **dadurch gekennzeichnet, daß** die Anfragestruktur ein Schreibflag aufweist zum Anzeigen, daß der Schreibvorgang auf die Platteneinrichtung ausgeführt wird.

12. Das Computersystem nach Anspruch 11, **dadurch gekennzeichnet, daß** das Plattenschreibmittel aufweist:

Mittel zum Detektieren, ob ein Schreibflag der Anfragestruktur gesetzt ist oder nicht;
Mittel zum Setzen des Schreibflags, falls das

Schreibflag nicht gesetzt ist;
Mittel zum Speichern von Schreibdaten der bestimmten Warteschlange auf der Platteneinrichtung, basierend auf einer Information der Anfragestruktur; und
Mittel zum Entfernen der Anfragestruktur und der zugehörigen Schreibdaten, wenn das Schreiben der Daten auf die Platteneinrichtung beendet ist.

13. Ein Verfahren zum Steuern eines Plattenzugriffs in einem Computersystem, das periodisch eine Prüfpunktoperation ausführt, wobei das System von dem letzten Prüfpunkt vor einem Fehler, der darin aufgetreten ist, neu startet, wobei das System eine Platteneinrichtung zum Speichern von in dem System zu verarbeitenden Daten und ein Betriebssystem zum Ausgeben einer Schreibanfrage an die Platteneinrichtung aufweist, **dadurch gekennzeichnet, daß** das Verfahren folgende Schritte aufweist:

Erstes Speichern (S10) von Schreibdaten und einer zugehörigen Schreibanfrage von dem genannten Betriebssystem an die genannte Platteneinrichtung in einem nichtflüchtigen Speicher (3);
zweites Speichern (S17, S130) der in dem nichtflüchtigen Speicher (3) gespeicherten Schreibdaten in die genannte Platteneinrichtung (4), wobei der zweite Speicherschritt den Schritt des Speicherns der vor dem letzten Prüfpunkt erzeugten Schreibdaten aus dem nichtflüchtigen Speicher (3) in die Platteneinrichtung (4) umfaßt.

14. Das Verfahren nach Anspruch 13, das ferner den Schritt des Meldens (S13) einer Beendigung des Schreibens an das Betriebssystem aufweist, nachdem die Schreibdaten und die zugehörige Schreibanfrage in den nichtflüchtigen Speicher geschrieben sind, wenn eine Schreibanfrage von dem Betriebssystem an die Platteneinrichtung ausgegeben wird.

15. Das Verfahren nach Anspruch 13, das ferner folgende Schritte aufweist:

Erstes Lesen (S21, S25) der Schreibdaten aus dem nichtflüchtigen Speicher, wenn erfaßte Schreibdaten unter den in dem nichtflüchtigen Speicher gespeicherten Schreibdaten sind, wenn das Betriebssystem eine Leseanfrage an die Platteneinrichtung ausgibt und
zweites Lesen (S21, S23) der Schreibdaten von der Platteneinrichtung, wenn unter den in dem nichtflüchtigen Speicher (3) gespeicherten Schreibdaten nicht erfaßte Schreibdaten sind.

16. Das Verfahren nach Anspruch 13, das ferner den Schritt des Ungültigmachens (S17) aller Schreibdaten und der korrespondierenden Schreibanfragen umfaßt, die nach dem letzten Prüfpunkt erzeugt worden sind, wenn ein Fehler aufgetreten ist.

17. Das Verfahren nach Anspruch 13, wobei der nichtflüchtige Speicher eine unbestimmte Warteschlange (31) zum Speichern von Schreibdaten und der zugehörigen Schreibanfrage, die nach dem letzten Prüfpunkt erzeugt wurden, und eine bestimmte Warteschlange (32) zum Speichern von Schreibdaten, die vor dem letzten Prüfpunkt erzeugt wurden, auf der Platteneinrichtung umfaßt, wobei das Verfahren ferner den Schritt des Verschiebens (S15) der Schreibanfrage und Schreibdaten, die in der unbestimmten Warteschlange gespeichert sind, in eine bestimmte Warteschlange aufweist, während eine Prüfpunktoperation ausgeführt wird.

18. Das Verfahren nach Anspruch 13, wobei der nichtflüchtige Speicher eine unbestimmte Warteschlange (31) zum Speichern von Schreibdaten und der zugehörigen Schreibanfrage, die nach dem letzten Prüfpunkt erzeugt wurde, und eine bestimmte Warteschlange (32) zum Speichern von Schreibdaten, die vor dem letzten Prüfpunkt erzeugt wurden, in der Platteneinrichtung aufweist, wobei der zweite Speicherschritt den Schritt des Speicherns (S17) der in der bestimmten Warteschlange gespeicherten Schreibdaten in der Platteneinrichtung umfaßt.

19. Das Verfahren nach Anspruch 13, wobei der nichtflüchtige Speicher eine unbestimmte Warteschlange (31) zum Speichern von Schreibdaten und der zugehörigen Schreibanfrage, die nach dem letzten Prüfpunkt erzeugt wurden, und eine bestimmte Warteschlange (32) zum Speichern von Schreibdaten und der zugehörigen Schreibanfrage, die vor dem letzten Prüfpunkt erzeugt wurden, in der Platteneinrichtung aufweist, wobei das Verfahren ferner den Schritt des Speicherns (S19) aller Schreibdaten, die in der bestimmten Warteschlange gespeichert sind, in die Platteneinrichtung aufweist, während sich das Computersystem wiederherstellt.

20. Das Verfahren nach Anspruch 13, wobei der nichtflüchtige Speicher eine unbestimmte Warteschlange (31) zum Speichern aller Schreibdaten und der zugehörigen Schreibanfrage, die nach dem letzten Prüfpunkt erzeugt wurden, aufweist, wobei der erste Speicherschritt den Schritt des Verwerfens (S17) der Schreibdaten und der Schreibanfragen in der unbesdtimmten Warteschlange nach dem Auftreten eines Fehlers aufweist.

Revendications

1. Système informatique exécutant périodiquement une opération de point de contrôle, le système redémarrant à partir du dernier point de contrôle avant une erreur survenant à l'intérieur, le système comprenant une unité de disque (4) pour stocker des données à traiter dans le système, et un système d'exploitation (15) pour délivrer une demande d'écriture à l'unité de disque, **caractérisé en ce que** ladite unité de disque sert à la lecture et le stockage de données ; ledit système d'exploitation sert à délivrer une demande d'écriture ou de lecture à l'unité de disque ; et **en ce que** le système comprend en outre :

   une mémoire non volatile (3) ;
   des moyens de stockage de demandes d'écriture (11) pour stocker des données d'écriture et une demande d'écriture associée dans la mémoire non volatile ; et
   des moyens d'écriture sur disque (14) pour stocker les données d'écriture stockées dans la mémoire non volatile par les moyens de stockage de demandes d'écriture dans l'unité de disque, les moyens d'écriture sur disque comprenant en outre des moyens pour stocker des données d'écriture selon la demande d'écriture générée avec le dernier point de contrôle de la mémoire non volatile à l'unité de disque.

2. Système informatique selon la revendication 1, **caractérisé en ce que** le système comprend des moyens pour notifier une fin d'écriture au système d'exploitation, après avoir stocké la demande d'écriture et les données d'écriture associées dans la mémoire non volatile, lorsqu'une demande d'écriture du système d'exploitation à l'unité de disque est délivrée.

3. Système informatique selon la revendication 1, **caractérisé en ce que** le système comprend en outre :

   des moyens pour lire les données d'écriture de la mémoire non volatile s'il y a les données correspondantes sur la mémoire non volatile, lorsque le système d'exploitation délivre une demande de lecture à l'unité de disque ; et
   des moyens pour lire les données d'écriture de l'unité de disque, s'il n'y a pas les données correspondantes dans la mémoire non volatile.

4. Système informatique selon la revendication 1, **caractérisé en ce que** le système comprend en outre des moyens de suppression de demandes d'écriture (13) pour invalider des données d'écriture et les demandes d'écriture correspondantes générées après le dernier point de contrôle, lorsqu'une erreur survient dans le système.

5. Système informatique selon la revendication 1, **caractérisé en ce que** les moyens de stockage de demandes d'écriture comprennent une file d'attente indéfinie (31) pour stocker une demande d'écriture qui est générée après le dernier point de contrôle, et une file d'attente définie (32) pour stocker une demande d'écriture qui est générée avant le dernier point de contrôle.

6. Système informatique selon la revendication 5, **caractérisé en ce que** le système comprend en outre des moyens de détermination de demandes d'écriture (12) pour transférer les données d'écriture stockées de la file d'attente indéfinie à la file d'attente définie, alors qu'une opération de point de contrôle est exécutée.

7. Système informatique selon la revendication 5, **caractérisé en ce que** les moyens de stockage de demandes d'écriture comprennent des moyens pour stocker la demande d'écriture et les données d'écriture associées dans la file d'attente indéfinie.

8. Système informatique selon la revendication 5, **caractérisé en ce que** les moyens d'écriture sur disque comprennent des moyens pour stocker les données d'écriture stockées dans la file d'attente définie dans l'unité de disque.

9. Système informatique selon la revendication 5, **caractérisé en ce que** le système comprend des moyens pour stocker toutes les données d'écriture stockées dans la file d'attente définie dans l'unité de disque, tout en redémarrant le système informatique.

10. Système informatique selon la revendication 5, **caractérisé en ce que** les files d'attente indéfinies et définies comprennent des moyens de blocs de contrôle pour contrôler chacune des données d'écriture et la demande d'écriture associée, chacune des données d'écriture ayant une structure de demande pour lier des données d'écriture et la structure de demande suivante.

11. Système informatique selon la revendication 10, **caractérisé en ce que** la structure de demande comprend un indicateur d'écriture pour indiquer que le traitement d'écriture sur l'unité de disque est exécuté.

12. Système informatique selon la revendication 11, **caractérisé en ce que** les moyens d'écriture sur disque comprennent :

des moyens pour détecter si un indicateur d'écriture de la structure de demande est initialisé ou non ;
des moyens pour initialiser l'indicateur d'écriture, si l'indicateur d'écriture n'est pas initialisé ;
des moyens pour stocker des données d'écriture de la file d'attente définie sur l'unité de disque, sur la base d'informations de la structure de demande ; et
des moyens pour supprimer la structure de demande et les données d'écriture associées lorsque l'écriture des données sur l'unité de disque est achevée.

13. Procédé de contrôle d'accès au disque dans un système informatique exécutant périodiquement une opération de point de contrôle, le système redémarrant à partir du dernier point de contrôle avant une erreur survenant à l'intérieur, le système comprenant une unité de disque pour stocker des données à traiter dans le système et un système d'exploitation pour délivrer une demande d'écriture sur l'unité de disque, **caractérisé par** les étapes de :

premier stockage (S10) des données d'écriture et une demande d'écriture associée dudit système d'exploitation à ladite unité de disque dans une mémoire non volatile (3) ;
deuxième stockage (S17, S130) des données d'écriture stockées dans la mémoire non volatile (3) dans ladite unité de disque (4), l'étape de deuxième stockage comprenant l'étape de stocker des données d'écriture générées avant le dernier point de contrôle de la mémoire non volatile (3) à l'unité de disque (4).

14. Procédé selon la revendication 13, comprenant en outre l'étape de notification (S13) d'une fin d'écriture au système d'exploitation, après avoir stocké les données d'écriture et la demande d'écriture associée dans la mémoire non volatile, lorsqu'une demande d'écriture du système d'exploitation à l'unité de disque est délivrée.

15. Procédé selon la revendication 13, comprenant en outre les étapes de :

première lecture (S21, S25) des données d'écriture de la mémoire non volatile, si des données d'écriture sont acquises parmi les données d'écriture stockées dans la mémoire non volatile, lorsque le système d'exploitation délivre une demande de lecture à l'unité de disque ; et
deuxième lecture (S21, S23) des données d'écriture de l'unité de disque, si des données d'écriture ne sont pas acquises parmi les données d'écriture stockées dans la mémoire non volatile (3).

16. Procédé selon la revendication 13, comprenant en outre l'étape d'invalidation (S17) de toutes les données d'écriture et de la demande d'écriture correspondante générées après le dernier point de contrôle, si une erreur est apparue

17. Procédé selon la revendication 13, dans lequel la mémoire non volatile comprend une file d'attente indéfinie (31) pour stocker des données d'écriture et la demande d'écriture associée générées après le dernier point de contrôle, et une file d'attente définie (32) pour stocker des données d'écriture dans l'unité de disque générées avant un dernier point de contrôle, comprenant en outre l'étape de transfert (S15) de la demande d'écriture et des données d'écriture stockées d'une file d'attente indéfinie à une file d'attente définie, pendant l'exécution de l'opération du point de contrôle.

18. Procédé selon la revendication 13, dans lequel la mémoire non volatile comprend une file d'attente indéfinie (31) pour stocker des données d'écriture et la demande d'écriture associée générées après le dernier point de contrôle, et une file d'attente définie (32) pour stocker des données d'écriture dans l'unité de disque générées avant le dernier point de contrôle, dans lequel l'étape de deuxième stockage comprend l'étape de stockage (S17) des données d'écriture stockées dans la file d'attente définie dans l'unité de disque.

19. Procédé selon la revendication 13, dans lequel la mémoire non volatile comprend une file d'attente indéfinie (31) pour stocker des données d'écriture et la demande d'écriture associée générées après le dernier point de contrôle, et une file d'attente définie (32) pour stocker des données d'écriture et la demande d'écriture associée dans l'unité de disque générées avant le dernier point de contrôle, comprenant en outre l'étape de stockage (S19) de toutes les données d'écriture stockées dans la file d'attente définie dans l'unité de disque, pendant la restauration du système informatique.

20. Procédé selon la revendication 13, dans lequel la mémoire non volatile comprend une file d'attente indéfinie (31) pour stocker des données d'écriture et la demande d'écriture associée générées après le dernier point de contrôle, dans lequel la première étape de stockage comprend l'étape de suppression (S17) des données d'écriture et des demandes d'écriture dans la file d'attente indéfinie après apparition d'une erreur.

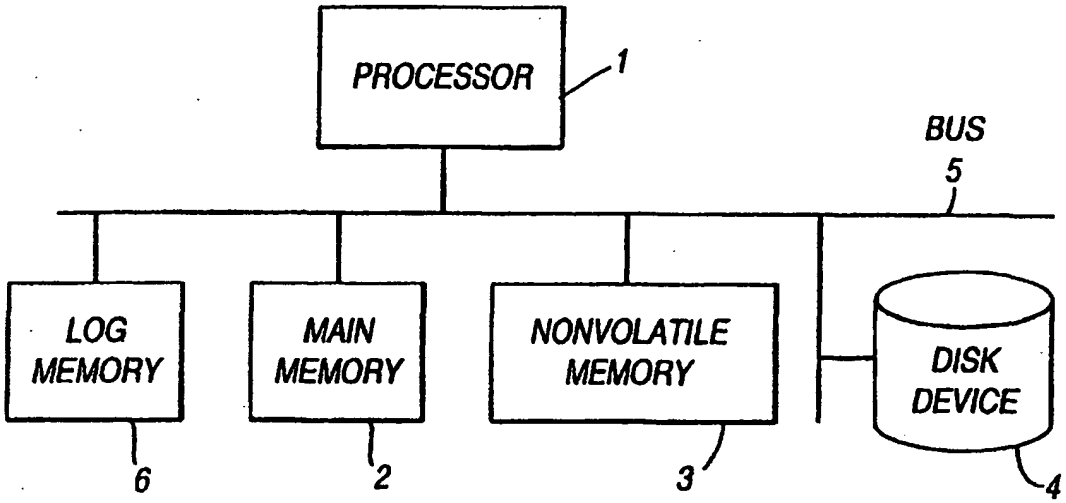
PROCESSOR
1
BUS
5
LOG MEMORY
6
MAIN MEMORY
2
NONVOLATILE MEMORY
3
DISK DEVICE
4

Fig. 1a

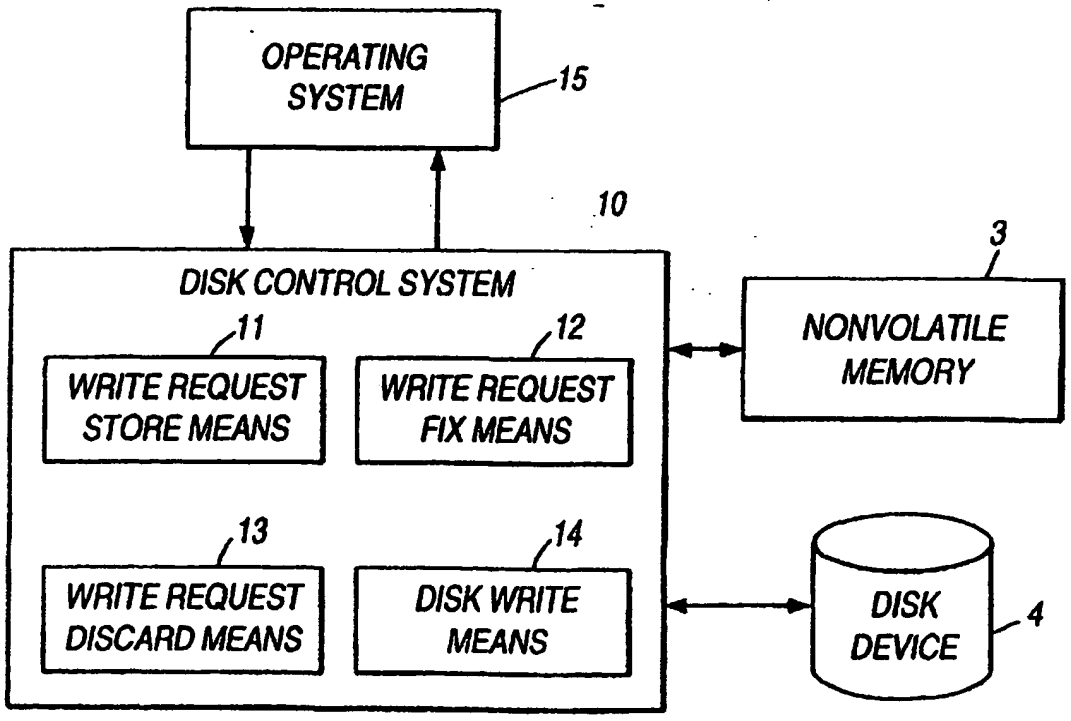
OPERATING SYSTEM
15
10
DISK CONTROL SYSTEM
11
WRITE REQUEST STORE MEANS
12
WRITE REQUEST FIX MEANS
13
WRITE REQUEST DISCARD MEANS
14
DISK WRITE MEANS
3
NONVOLATILE MEMORY
DISK DEVICE
4

Fig. 1b

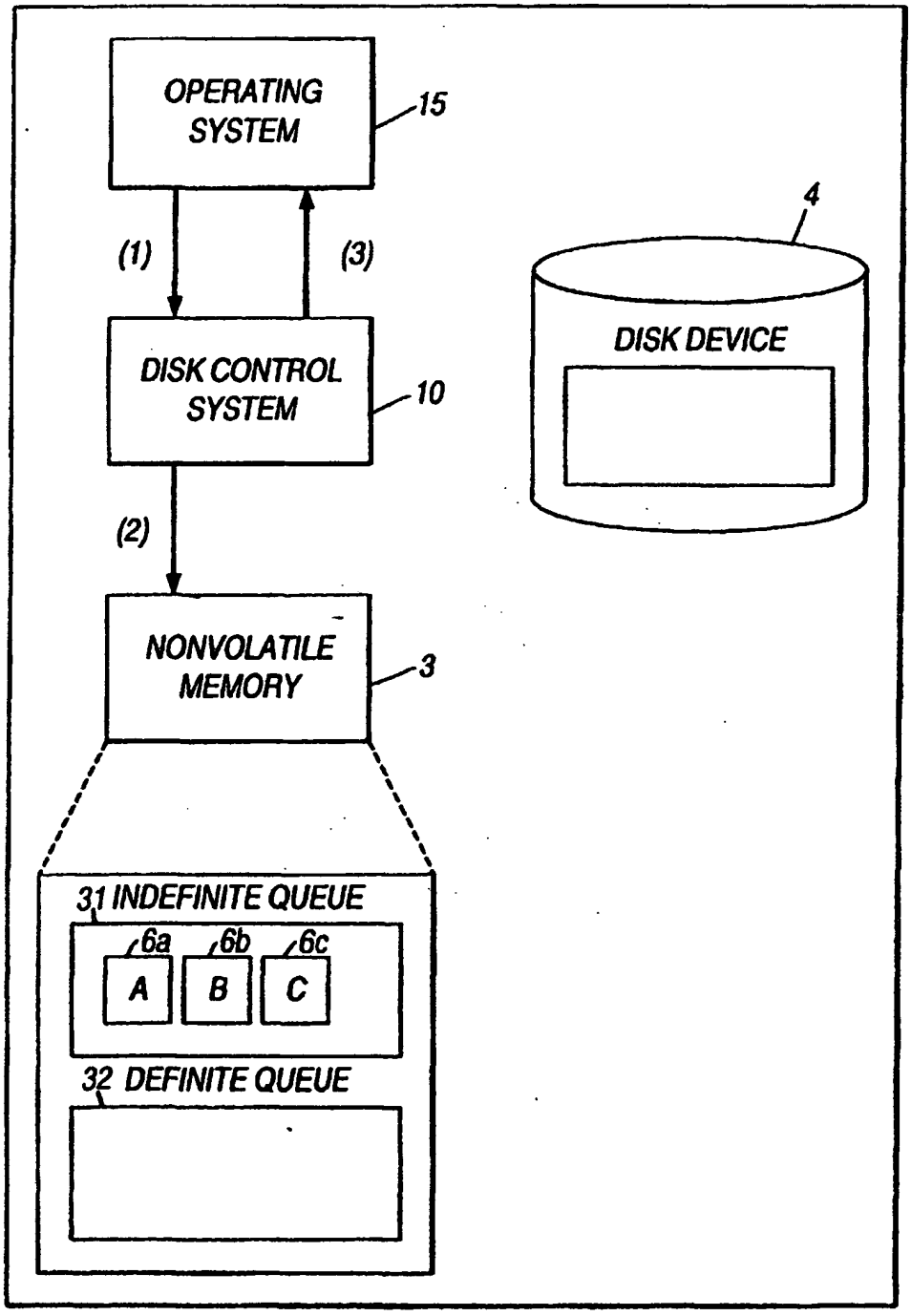
OPERATING SYSTEM
15
(1)
(3)
DISK CONTROL SYSTEM
10
(2)
NONVOLATILE MEMORY
3
4
DISK DEVICE
31 INDEFINITE QUEUE
6a
6b
6c
A
B
C
32 DEFINITE QUEUE

Fig. 2

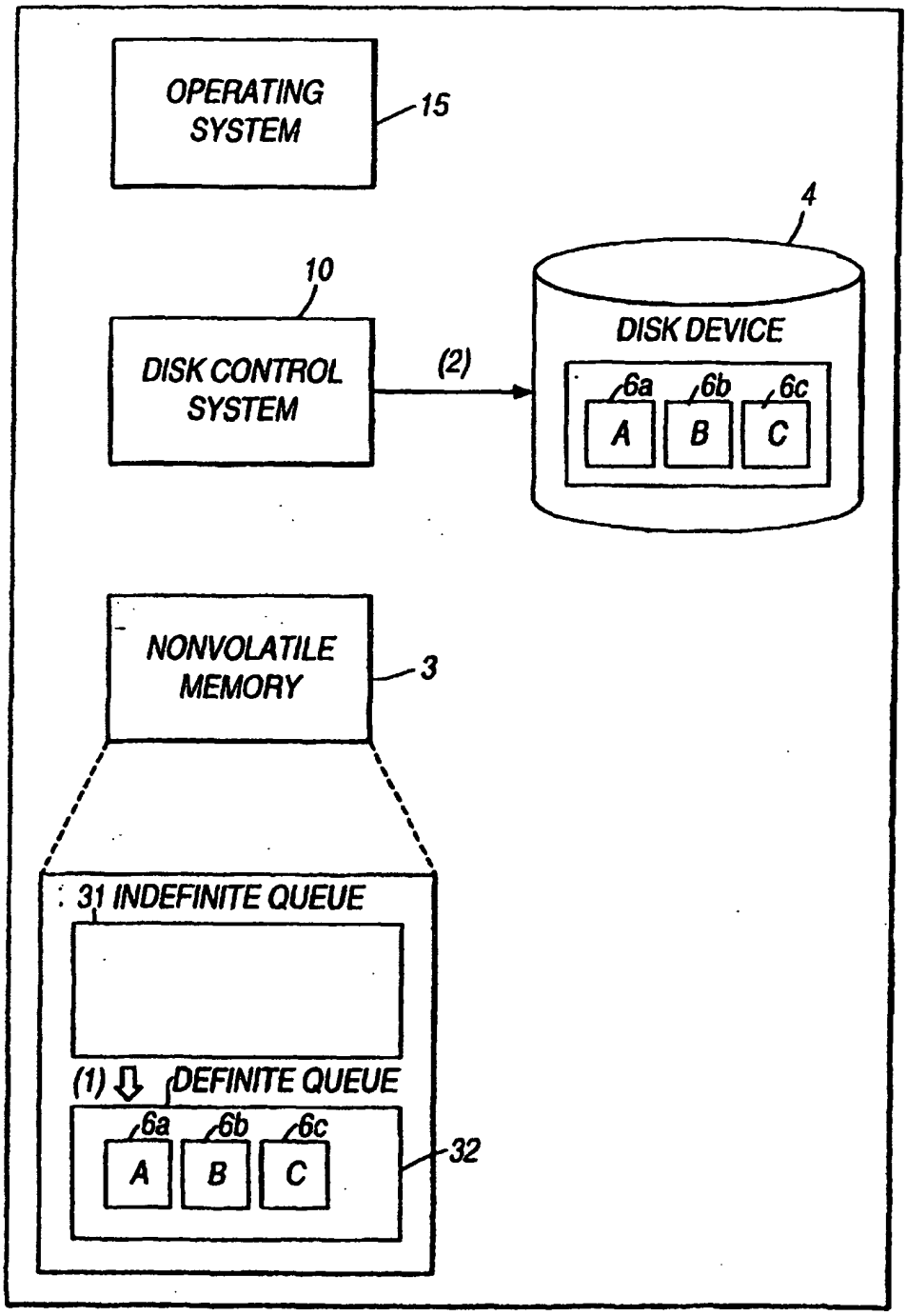
OPERATING SYSTEM
15
4
10
DISK CONTROL SYSTEM
(2)
DISK DEVICE
6a
6b
6c
A
B
C
NONVOLATILE MEMORY
3
31 INDEFINITE QUEUE
(1)
DEFINITE QUEUE
6a
6b
6c
A
B
C
32

Fig. 3

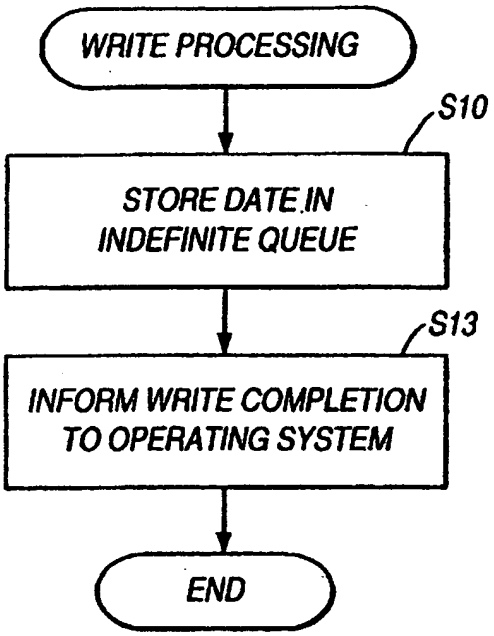

Fig. 4a

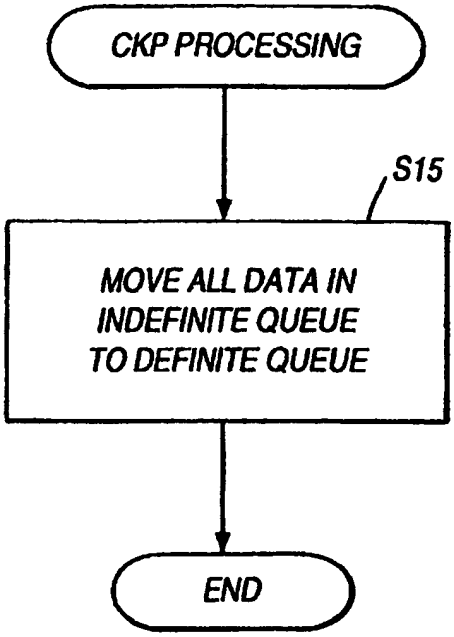

Fig. 4b

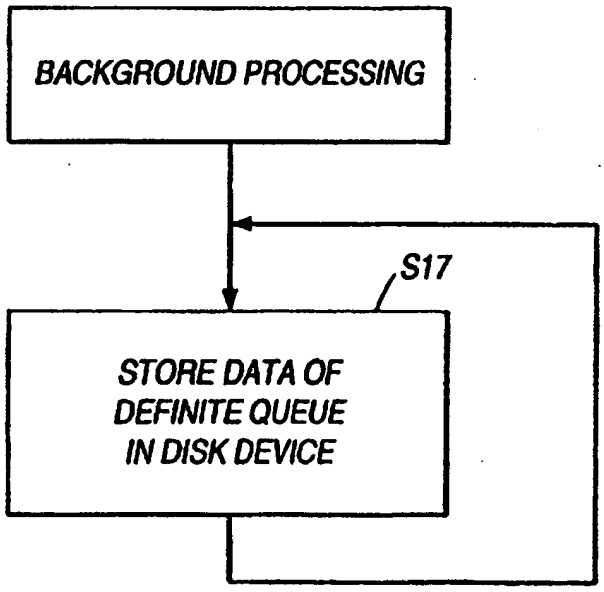

Fig. 4c

NONVOLATILE MEMORY STRUCTURE

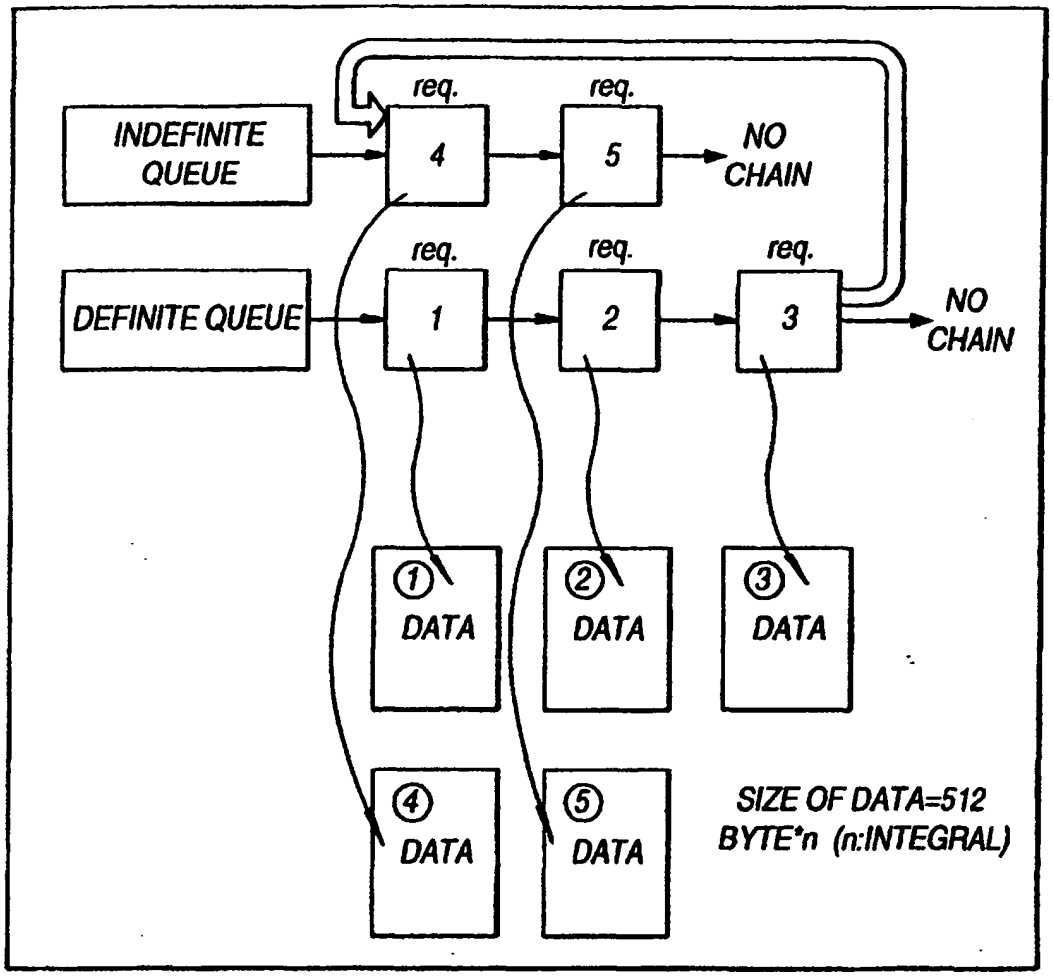

Fig. 5a

REQUEST STRUCTURE

DISK : N
SECTOR : M
NUMBER OF SECTOR : L
POINTER TO DATA
POINTER TO NEXT REQUEST
WRITE FLAG
(INDICATING WRITING PROCESS TO DISK)

Fig. 5c

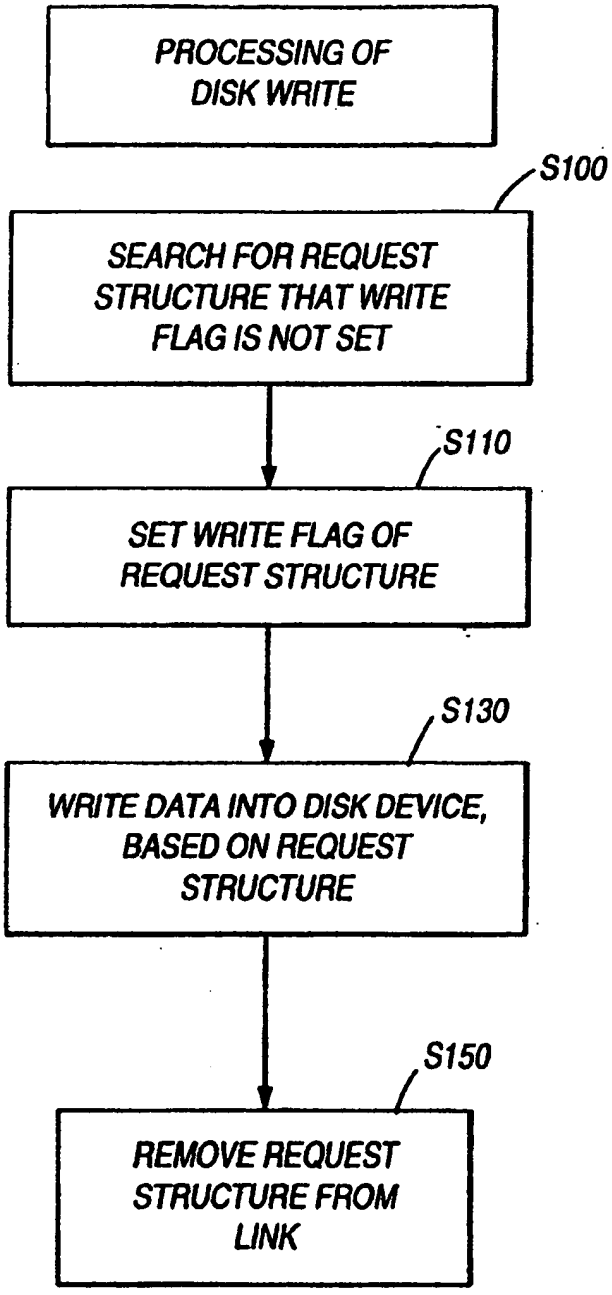
PROCESSING OF DISK WRITE
S100
SEARCH FOR REQUEST STRUCTURE THAT WRITE FLAG IS NOT SET
S110
SET WRITE FLAG OF REQUEST STRUCTURE
S130
WRITE DATA INTO DISK DEVICE, BASED ON REQUEST STRUCTURE
S150
REMOVE REQUEST STRUCTURE FROM LINK

Fig. 5b

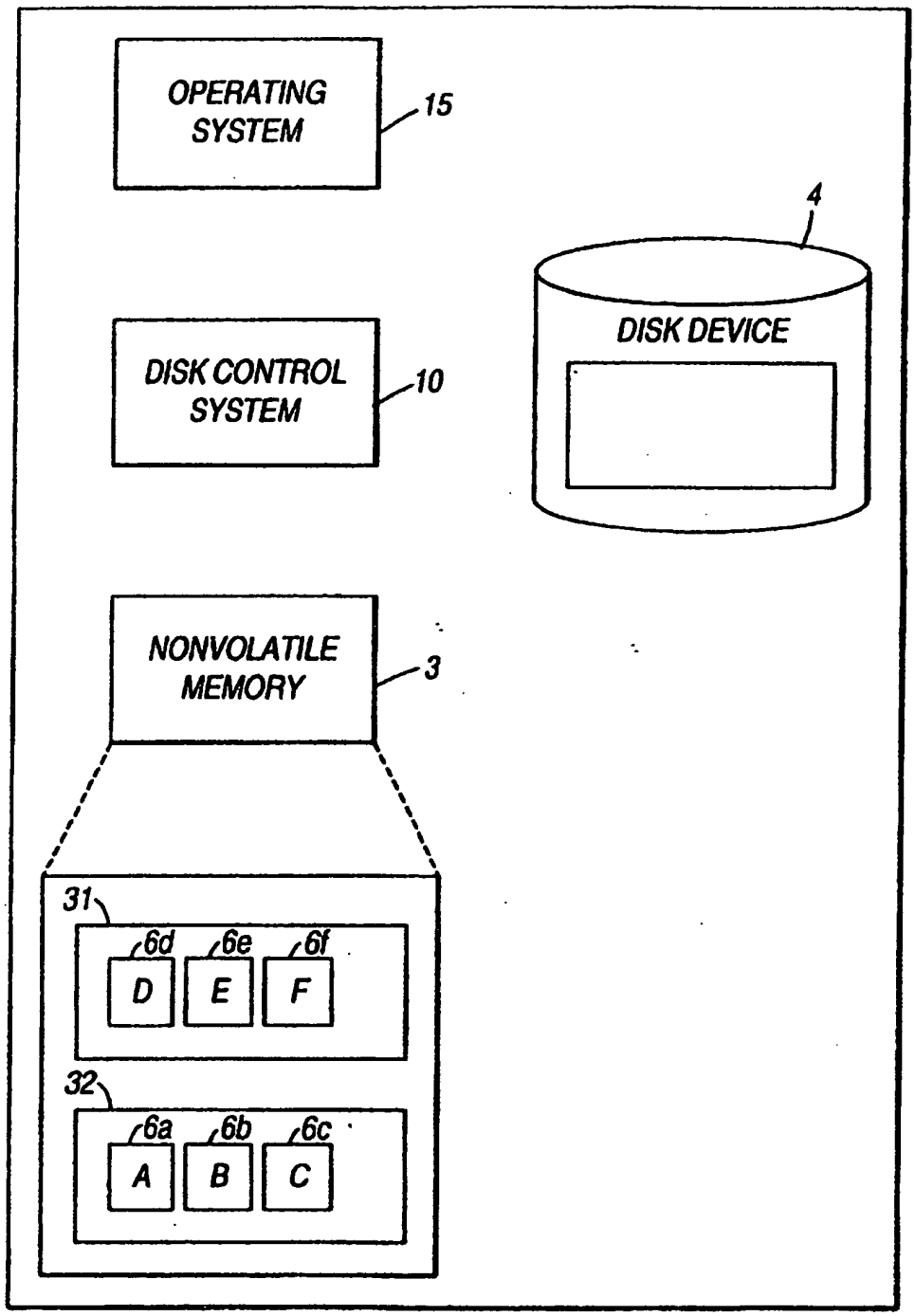
OPERATING SYSTEM
15
4
DISK DEVICE
DISK CONTROL SYSTEM
10
NONVOLATILE MEMORY
3
31
6d
6e
6f
D
E
F
32
6a
6b
6c
A
B
C

Fig. 6

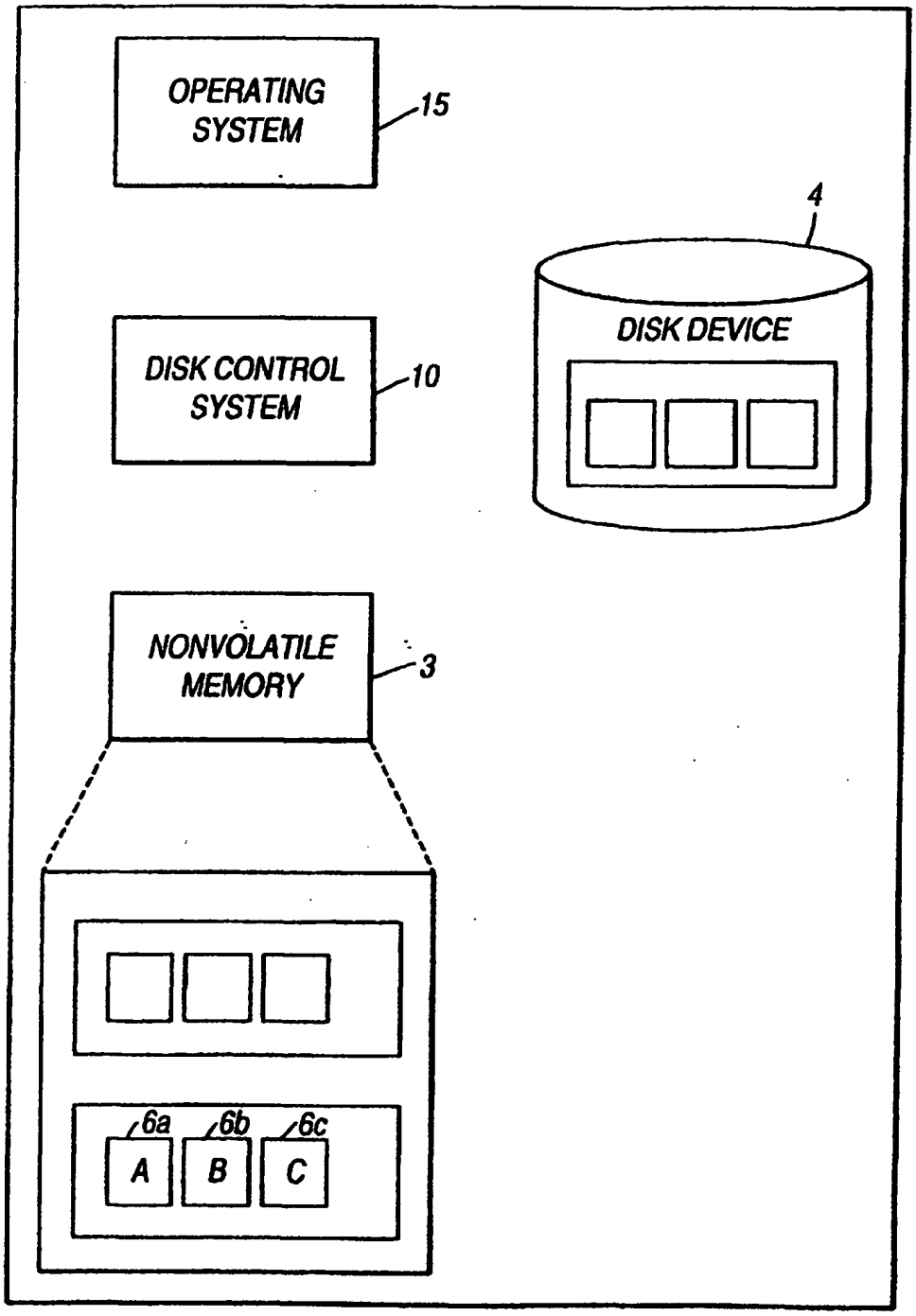
OPERATING SYSTEM
15
4
DISK DEVICE
DISK CONTROL SYSTEM
10
NONVOLATILE MEMORY
3
6a
6b
6c
A
B
C

Fig. 7

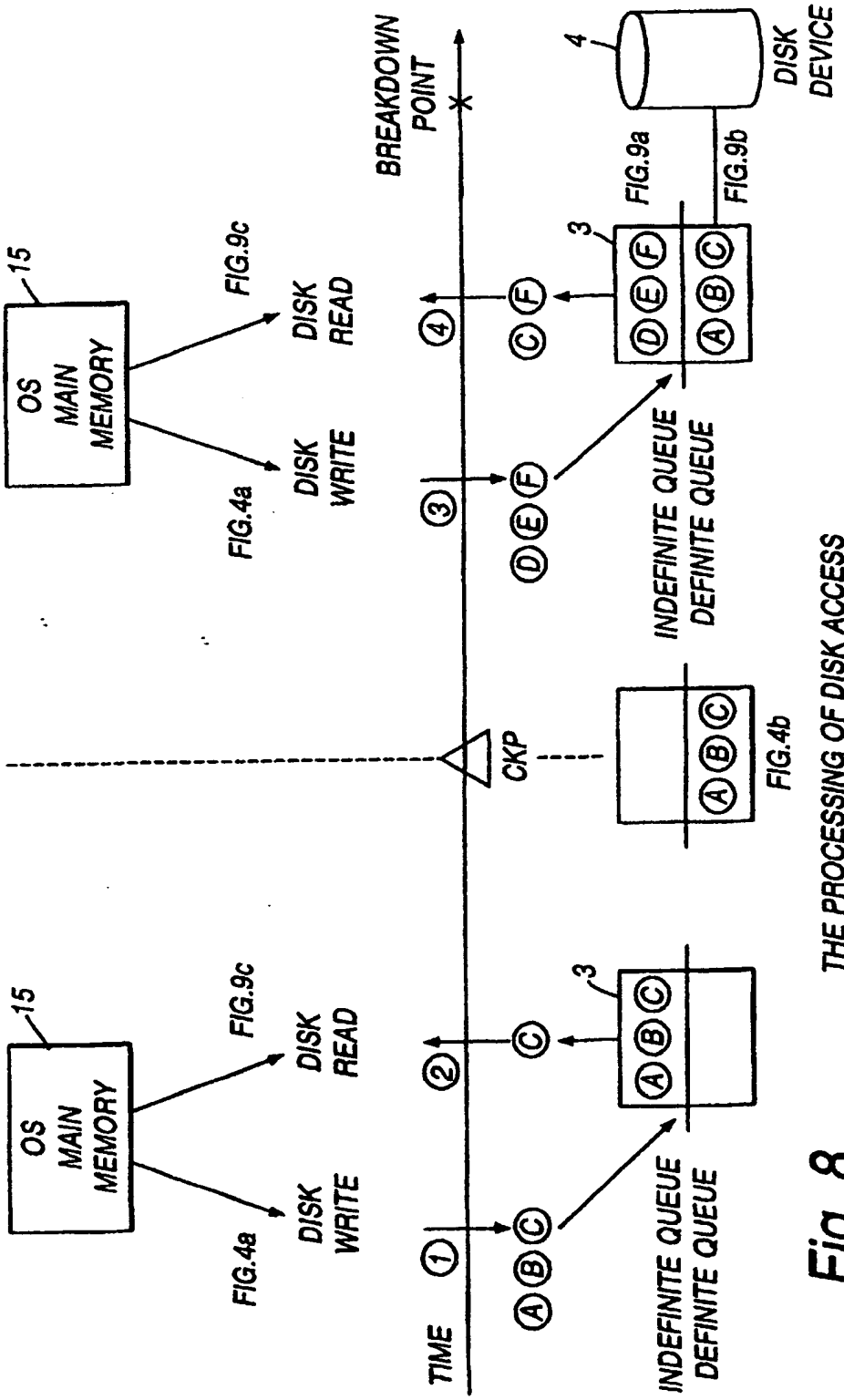

Fig. 8

THE PROCESSING OF DISK ACCESS REQUEST BEFORE/AFTER THE CHECKPOINT

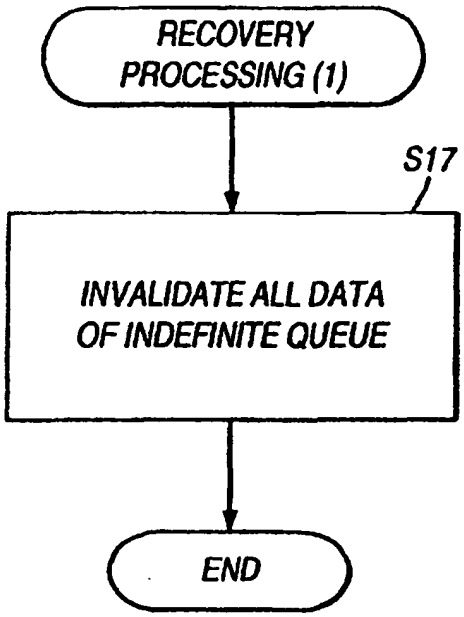

Fig. 9a

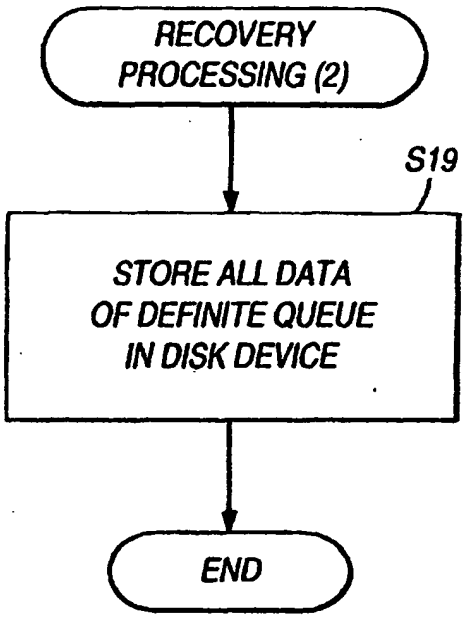

Fig. 9b

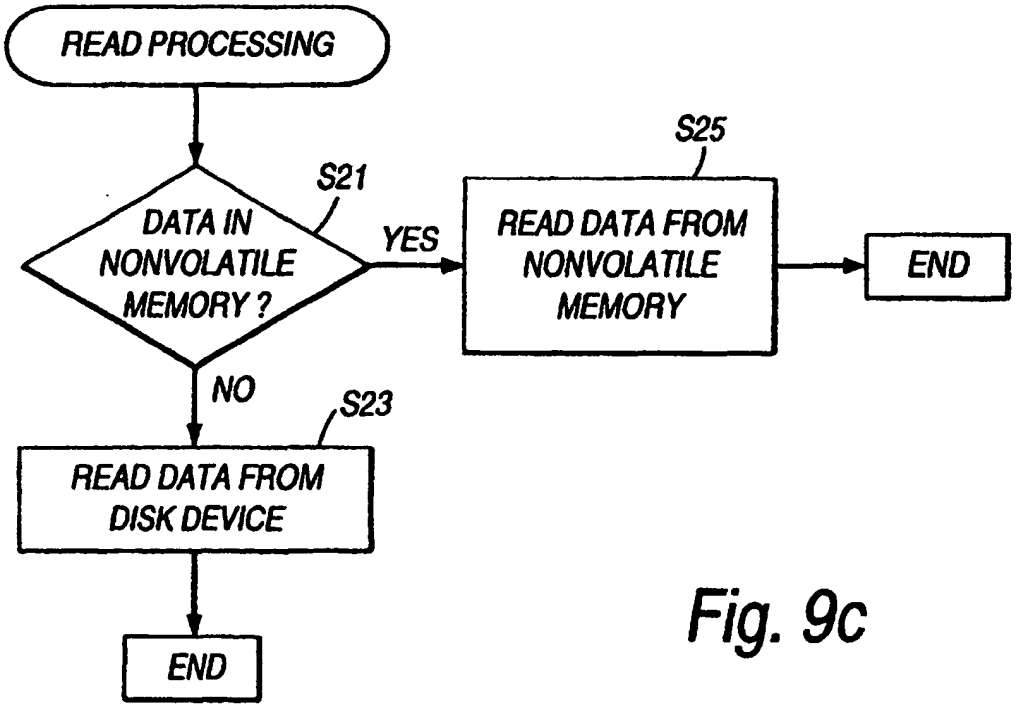

Fig. 9c